Spring Boot实际应用讲解（二）： 配置详解


ZYRzyr (/u/f8ff63b17fc7) + 关注
2017.11.14 11:02* 字数 1148 阅读 452 评论 0 喜欢 3
(/u/f8ff63b17fc7)

> 文/ZYRzyr (https://www.jianshu.com/u/f8ff63b17fc7)
> 原文链接:http://www.jianshu.com/p/d4c7f33c9b37
> (https://www.jianshu.com/p/d4c7f33c9b37)

本文提纲

一、默认配置
二、自定义配置（属性）
三、适配多环境
四、最后

本文运行环境

```
Ubuntu 16.04 LTS
JDK 8 +
IntelliJ IDEA ULTIMATE 2017.2
Maven 3.5.0
Spring Boot 1.5.8.RELEASE
```

一、默认配置

`Spring Boot` 提供了很多默认配置，不需要进行显示声明，并且大多数的默认配置已经能满足实际的开发，节省了很多时间。
通常一个 `moudle` 中的配置文件为： `application.yml` 或 `application.properties` ，里面可以声明如数据库连接、tomcat、缓存等等非常多的配置，可根据项目实际需要进行配置。 `.yml` 与 `.properties` 区别仅在于 `.yml` 使用 `YAML` 语法进行书写，更方便人阅读，所以本系列文章，均使用 `.yml` 作为后缀的配置文件。
某些情况下，需要改变配置参数，比如项目运行时进行配置参数修改，需要遵循以下配置优先级：

1. 命令行参数
2. 来自java:comp/env的JNDI属性
3. Java系统属性（System.getProperties()）
4. 操作系统环境变量
5. RandomValuePropertySource 属性类生成的 random.* 属性
6. 应用以外（jar包外部）的 application.yml（或 properties）（带spring.profile）文件
7. 打包在应用内（jar包内部）的 application.yml（或 properties）（带spring.profile）文件

8. 应用以外（jar包外部）的 application.yml（或 properties）（不带spring.profile）文件
9. 打包在应用内（jar包内部）的 application.yml（或 properties）（不带spring.profile）文件
10. @Configuration注解类上的@PropertySource
11. SpringApplication.setDefaultProperties 声明的默认属性

由于涉及内容众多，本系列暂不展开细讲。

二、自定义配置（属性）

2.1 定义属性类

一本书的属性包含：书名、价格、页数等，定义一个 `Book` 类，如下：

```
package com.zyr.demo.domain;

import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.stereotype.Component;

@Component  //将该类作为Bean注入Spring容器
@ConfigurationProperties(prefix = "book")  //配置文件中user前缀的属性将自动绑定到本类对应的
public class Book {
    private String name;
    private Double price;
    private Integer page;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Double getPrice() {
        return price;
    }

    public void setPrice(Double price) {
        this.price = price;
    }

    public Integer getPage() {
        return page;
    }

    public void setPage(Integer page) {
        this.page = page;
    }

    @Override
    public String toString() {
        return "Book{" +
                "name='" + name + '\'' +
                ", price=" + price +
                ", page=" + page +
                '}';
    }
}
```

2.2 添加配置参数

类定义完成后，就可在 `application.yml` 以 `key-value` 的形式，增加想自定义的配置，如下：

```
#Book.java的属性
book:
  name: 一本书
  price: 35.5
  page: 128
```

配置完成后，当使用 Book 类的时候，其字段将自动赋值配置中的值。

2.3 测试

编写 Book 的测试类，如下：

```
package com.zyr.demo.domain;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;

import static org.junit.Assert.assertEquals;

@RunWith(SpringRunner.class)
@SpringBootTest
public class BookTest {

    @Autowired
    private Book book;

    @Test
    public void test() {
        assertEquals("一本书", book.getName());
        assertEquals(35.5, book.getPrice(), .001);
        assertEquals(128, book.getPage(), .001);
    }
}
```

运行测试用例，测试通过，说明上面的自定义配置成功。

2.4 另外

Spring Boot 的 RandomValuePropertySource 类，提供了很多生成随机数的工具，如随机字符串，随机整数，指定范围内的随机数等等。
直接将 application.yml 中 book 的参数修改如下：

```
#Book.java的属性
book:
  name: ${random.value} #赋值随机字符串
  price: ${random.int}  #赋值随机整数
  page: ${random.int}   #赋值随机整数
```

将 BookTest.test 改为打印输出 Book ， 多次运行，控制台每次显示的 Book 的值都不一样：

```
@Test
public void test() {
    System.out.println(book);
}
```

三、适配多环境

通常在一个实际的项目中，会有开发环境，生产环境等，并且在不同的环境中，对应的配置如：数据库配置，端口号配置，缓存配置等，一般会不同，所以需要针对不同的环境，编写不同的配置文件，公用的配置直接写在 `application.yml` 中。

3.1 新增配置文件

在与 `application.yml` 同级的地方，即 `resources` 文件夹下新增如下两个配置文件：

```
application-dev.yml        //开发环境配置文件
application-pro.yml        //生成环境配置文件
```

3.2 使用指定环境的配置文件

通过在 `application.yml` 使用 `spring.profiles.active` 配置，即可使用指定环境的配置文件，如：

```
#文件`application.yml`

spring:
  profiles:
    active: dev  #使用开发环境配置
```

```
文件`application-dev.yml`

#Book.java的属性
book:
  name: ${random.value} #赋值随机字符串
  price: ${random.int}  #赋值随机整数
  page: ${random.int}   #赋值随机整数
```

```
文件`application-pro.yml`

#Book.java的属性
book:
  name: 一本书
  price: 35.5
  page: 128
```

3.3 测试

现在是开发环境，运行上面的测试用例 `BookTest.test` ，控制台输出随机数的 `Book` ：

```
Book{name='d326d313d13023c9b32f9f76efe90063', price=1.8404973E9, page=1297575634}
```

将 `application.yml` 改为：

```
spring:
  profiles:
    active: pro #使用生成环境配置
```

再次运行测试用例 `BookTest.test` ，控制台输出 `application-pro.yml` 中指定的 `Book` 的值：

```
Book{name='一本书', price=35.5, page=128}
```

最后

本文介绍了实际开发中 `Spring Boot` 常用的配置情况，关于 `application.yml` 中详细的配置说明，请参考官方说明 (https://link.jianshu.com?t=https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html)。

本文代码已上传至我的GitHub仓库 (https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo)，进入以后将branches (https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo/branches)切换为2-Config (https://link.jianshu.com?t=https://github.com/ZYRzyr/SpringBootDemo/tree/2-Config)即可看见。

前篇：
Spring Boot实际应用讲解（一）：Hello World
(https://www.jianshu.com/p/60f7e025c680)

后续将推出以下文章，敬请关注！

Spring Boot实际应用讲解（三）：表单验证 (https://www.jianshu.com/p/a2b4e61b5532)
Spring Boot实际应用讲解（四）：RESTful API
(https://www.jianshu.com/p/e907595e9d1d)
Spring Boot实际应用讲解（五）：AOP之请求日志
(https://www.jianshu.com/p/93216bf41182)
Spring Boot实际应用讲解（六）：MySQL + Spring-data-jpa(Hibernate)
(https://www.jianshu.com/p/b204472d8126)
Spring Boot实际应用讲解（七）：统一异常处理
Spring Boot实际应用讲解（八）：MySQL + Mybatis
Spring Boot实际应用讲解（九）：MySQL + Mybatis + Redis

文中若有错之处，还请各位批评指正，谢谢！

> 原文作者/ZYRzyr (https://www.jianshu.com/u/f8ff63b17fc7)
> 原文链接:http://www.jianshu.com/p/d4c7f33c9b37
> (https://www.jianshu.com/p/d4c7f33c9b37)

Spring Boot (/nb/18796030)　　举报文章　© 著作权归作者所有

ZYRzyr (/u/f8ff63b17fc7) ♂
写了 17755 字，被 32 人关注，获得了 92 个喜欢
(/u/f8ff63b17fc7)

+关注

程序猿一枚 技能树加点情况： 移动端—Android——5/5 移动端—iOS——1/5 前端—JavaScript、HTML—…

喜欢的老铁，来一波关注666

赞赏支持